Sentiment Analysis on Twitter Data

We have performed the text mining on the twitter data set.

Packages:

For this demonstration, we need to use these packages.

```
# read in the libraries
library(tidyverse) # general utility & workflow functions
library(tidytext) # tidy implimentation of NLP methods
library(topicmodels) # for LDA topic modelling
library(tm) # general text mining functions, making document term matri
library(SnowballC) # for stemming
library(NLP)
library (dplyr)
library (ggplot2)
library(RColorBrewer)
library (wordcloud)
```

Read & Show Dataset:

```
# read in our data
twitter_data <- read_csv(file.choose())

twitter_review <- c(twitter_data$review,twitter_data$rating) #only store

view(twitter_review)

head(twitter_review,10) #see the stored first 10 data

typeof(twitter_review)  #verify the data type
```

This read the dataset and show the head rows of the dataset.

Counting & Distribution:

Here is the code for counting words, length of words, and punctuations.

```
user_reviews <- twitter_data %>%
  mutate(length = str_length(review),
         npunct = str_count(review, "[[:punct:]]" ),
         nword = str_count(review, "\\w+" ))
head(user_reviews,20)
```

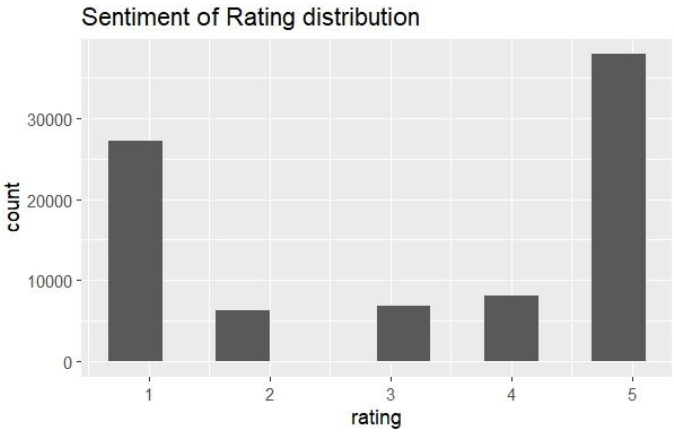
Output:

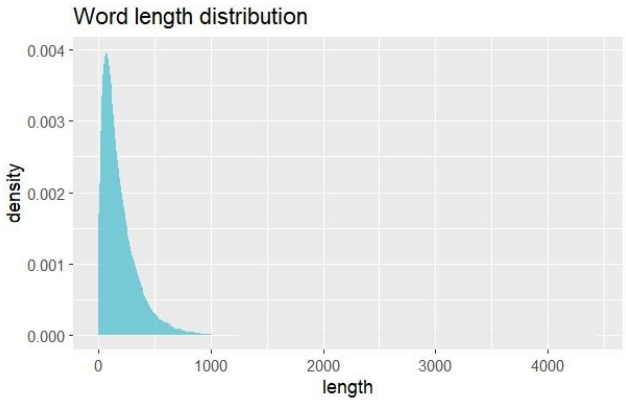
```
> head(user_reviews,20)
# A tibble: 20 × 6
   review                                rating comments length npunct nwo
rd
   <chr>                                  <dbl> <chr>     <int>  <int> <in
t>
 1 "Wonderfully pretty app with everything…      5 Astonis…    258      6
37
 2 "Buy it, the best twitter app just got …      5 Simply …     50      2
10
 3 "Best Twitter client out there made eve…      5 The Bes…     59      2
9
 4 "Tweetie 2.0 fixes the few things that …      5 King of…    109      3
```

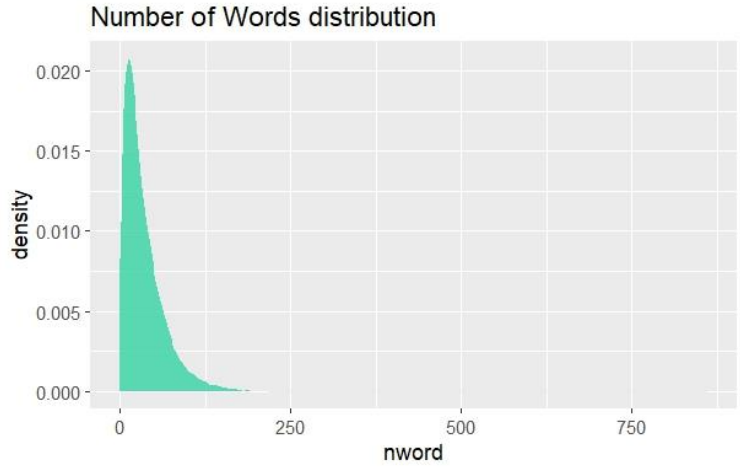
Then we have to check the sentiment of rating distribution, length distribution and number of word distribution.

```
# checking the distribution
user_reviews%>%
  ggplot( aes(x=rating)) +
  ggtitle( "Sentiment of Rating distribution" ) +
  geom_histogram(bins = 10 )

user_reviews %>%ggplot( aes(x=length)) +
  ggtitle( "Word length distribution" ) +
  geom_density(fill= "#59c3d2" , color= "#e9ecef" , alpha= 0.8 )
user_reviews %>%
  ggplot( aes(x=nword)) +
  ggtitle( "Number of Words distribution" ) +
  geom_density(fill= "#36d3a2" , color= "#e9ecef" , alpha= 0.8 )
```







From distribution, we can see that frequency of sentiment value 5 is really high. Also, most of the phrases have lengths between 0 to 1000. And the majority of the text has less than 125 words.

Tokenization:

One popular task in Natural Language Processing (NLP) is tokenization. "Tokens" are usually individual words and "tokenization" are taking a text or set of document/text and splitting it up into individual words.

```
#tokenization
review_Tokens <- user_reviews %>%
  unnest_tokens(word,review)

head(review_Tokens, 10 )
```

```
> head(review_Tokens, 10 )
# A tibble: 10 × 6
   rating comments       length npunct nword word
    <dbl> <chr>             <int>  <int> <int> <chr>
 1      5 Astonishing...     258      6    37 wonderfully
 2      5 Astonishing...     258      6    37 pretty
 3      5 Astonishing...     258      6    37 app
 4      5 Astonishing...     258      6    37 with
 5      5 Astonishing...     258      6    37 everything
 6      5 Astonishing...     258      6    37 you
 7      5 Astonishing...     258      6    37 could
 8      5 Astonishing...     258      6    37 possibly
```

Stop Words & Top Words:

In computing, stop words are words that are filtered out before or after the natural language data (text) are processed. While “ stop words ” typically refers to the most common words in a language.

List of some stop words added below as a screenshot from R Studio.

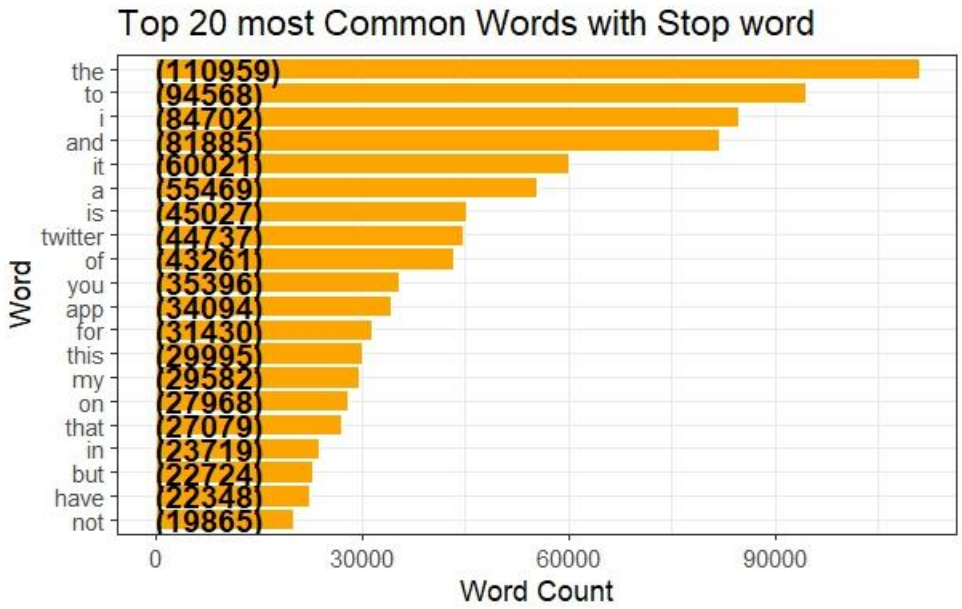
```
> print(stop_words)
# A tibble: 1,149 × 2
   word        lexicon
   <chr>       <chr>
 1 a           SMART
 2 a's         SMART
 3 able        SMART
 4 about       SMART
 5 above       SMART
 6 according   SMART
 7 accordingly SMART
 8 across      SMART
 9 actually    SMART
```

Top words before using stop words

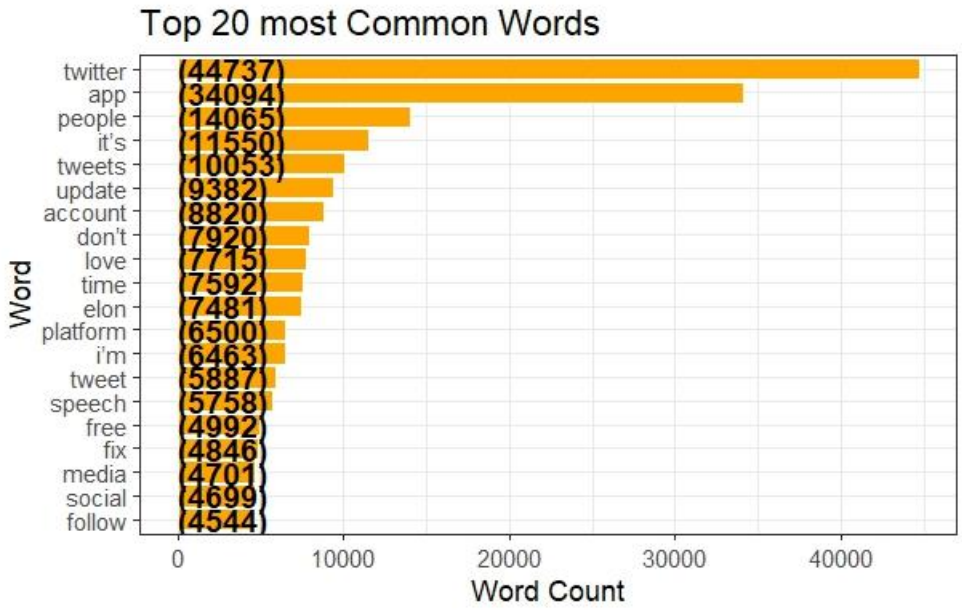
```r
# before removing stop words
user_reviews %>%
  unnest_tokens(word, review) %>%
  dplyr:: count(word,sort = TRUE ) %>%
  ungroup() %>%
  mutate(word = factor(word, levels = rev(unique(word)))) %>%
  head( 20 ) %>%
  ggplot(aes(x = word,y = n)) +
  geom_bar(stat= 'identity' ,colour= "white", fill= "orange") +
  geom_text(aes(x = word, y = 1 , label = paste0( "(" ,n, ")" ,sep= "" ))
            hjust= 0 , vjust= .5 , size = 4 , colour = 'black' ,
            fontface = 'bold' ) +
  labs(x = 'Word' , y = 'Word Count' ,
       title = 'Top 20 most Common Words with Stop word' ) +
  coord_flip() +
  theme_bw()
```

Output is :

Plot before removing stop words.



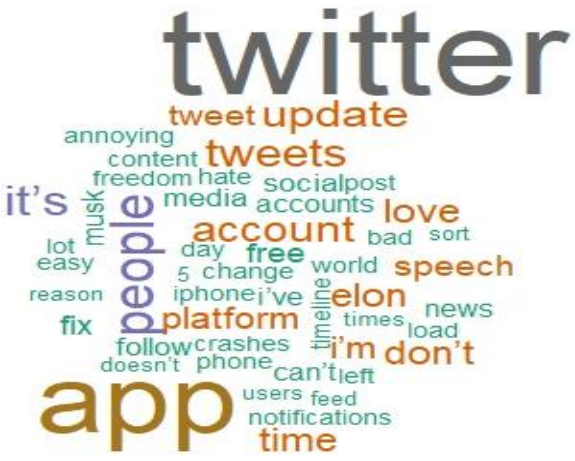
Plot after removing stop words



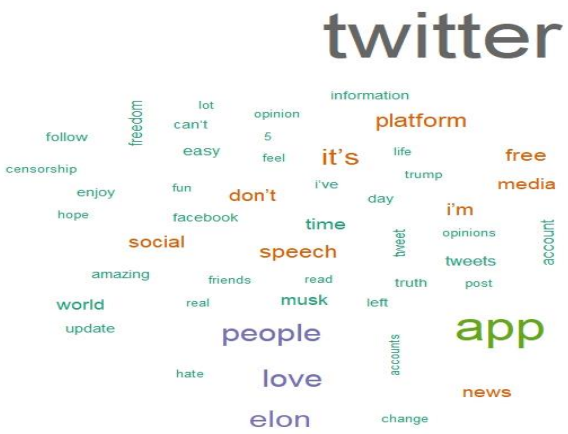
Word Cloud:

Word cloud is one of the common text data visualization tools. Here is word cloud for all data, then for sentiment values 5 and 2 to see the difference.

The plot of all data.



The plot of data with Sentiment value 5



The plot of data with sentiment value 2

Conclusion:

From this text data mining, we observed a lot of things. For example, from word cloud, we can

see the user review with low sentiment value has words like “annoying”, “fix”, “crashes”, ‘notification”. Which have a correlation with privacy update.